Extending Applications of Layer Selecting Rank Reduction

Stanford CS224N Default Project

Abraham Alappat
Department of Computer Science
Stanford University
abemdxb@stanford.edu

Abstract

Improving LLMs post training is gaining prominence due to training costs. In this study, we explore applying Layer Selective Rank Reduction ('LASER') - which reduces ranks of select weight matrices in a model - to drive performance improvement of a Multi-Task BERT model on sentiment analysis, paraphrase extraction and semantic textual similarity. Using versions of task-specific and multitask BERT models as baselines, we investigated performance impacts of LASER across tasks. We find that LASER can improve performance on certain tasks (like sentiment analysis), combining results of different LASER interventions is hard due to interlayer interactions. Our best results come from individual laser interventions in intermediate layers of the model.

1 Key Information to include

- Mentor:Andrew Hyungmin Lee
- External Collaborators (if you have any): N/A
- Sharing project: N/A

2 Introduction

Current state-of-the-art large language models are large and over-parameterized (Hinton et al., 2015), and expensive to train (Besiroglu et al., 2024). There has been concerted research into reducing model size while preserving performance, and into driving performance after pretraining. This includes research on model compression techniques like pruning (Zhu et al., 2023), low-rank adaptation (Hu et al., 2021; Yu et al., 2017), distillation (Hinton et al., 2015), low-rank training (Yang et al., 2020), and quantization, among others. Recent research indicates that model performance can be preserved with up to 90 percent size reduction (Frankle and Carbin, 2018), while another more recent paper (Sharma et al., 2024) has shown performance across large models like GPT-J and LlaMa 2 improving when certain model weights are reduced or pruned. The practice of reducing weights for given model however is largely unknown and is the focus of this initiative.

In this research, we explore the performance impact of Layer Specific Rank Reduction (LASER) across a multitask-finetuned BERT model trained across three narrow tasks; sentence sentiment classification, paraphrase detection and semantic textual similarity. We first fine-tuned task-specific baseline models on top of pretrained embeddings from BERT-base-uncased (Devlin et al., 2018). We then apply multitask learning (MTL) procedures to derive a higher performance model and enable testing of LASER on models enhanced with common MTL techniques. We use two versions of gradient surgery and annealed sampling (Stickland and Murray, 2019). Following this we experimented with different batch sizes and different levels of L2 regularization to moderate success. Finally, we further enhanced model architecture around the three task specific heads by including additional linear

layers, non-linear layers, and implementing cosine similarity for the STS task. We then apply LASER on individual MLP layer weights across the final multi-task model MB-COMB through two methods; drop-out and singular value decomposition to measure performance impact. Our findings reveal significant impact from individual LASER interventions. Given the low computational complexity and the tendency for the impact to center around certain key layers the we find the method promising for low cost enhancement.

Lastly we trial a layer-level greedy search method to find an optimal combination of LASER interventions. We find that gird-search of each layer and greedily adding interventions between layers did not extend performance beyond individual-layer level interventions. Our findings suggest that for BERT models, the most impactful interventions occur at intermediate layers, and that layer-interactions may result in greedy search starting at the end layer discovering local optima. This paper continues with a short review of related work in Section 3; and model architecture approach in section 4. We then perform a deep analysis of our experiments, including methods, details and results in section 5. Lastly we go through a qualitative analysis and provide a conclusion where we suggest future work around searching for optimal LASER interventions in Sections 6 and 7.

3 Related Work

In developing post-training methods for enhancing model performance, the goal is often preserve existing language representations while driving cost-savings. Previous efforts on model compression (pruning, quantization, weight sharing etc.) and model distillation have been largely applied with the goal of resource-savings (Bondarenko et al., 2021; Kwon et al., 2022; Touvron et al., 2021) and have generally indicated performance maintenance rather than performance improvement. There has been limited exploration of altering weights for the goal of enhancing performance despite new research around layer-specific knowledge representation (Conneau et al., 2020), layer specific entity knowledge (Meng et al., 2023), and the utility of intermediate layer representations to get the correct output (**?**).

A recent paper by Sharma et al. (2024) indicates that selective pruning or rank-reduction of layers in a model can actually drive up performance on Question-Answering tasks by removing conflicting representations from key layers. They propose a powerful technique called Layer Selective Rank Reduction which applies either Singular Value Decomposition or pruning to select layers across the model. They findings indicate significant uplift on QA tasks using GPT-J, LLaMA 2 models on HotPotQA. Through careful analysis of intermediate-layer embeddings from GPT-J, they show that for certain QA tasks that the model gets wrong, the higher-order weights store information of the same semantic-type as the right answer (e.g. "French" instead of "Dutch") and that the inclusion of both high-order and low-order weights averages the resulting embedding to generate common words like "the". Eliminating the higher-ordered weights enabled the right information to surface. Work on LASER has been limited to larger NLP models, and LASER's compatibility with other common model-enhancement techniques remain unanswered. Additionally, the development of an optimal combination of LASER interventions seem predicated on performing an expensive grid-search to find the most impactful individual intervention first. This motivated us to investigate the re-applicability of LASER to non-generative tasks on modified models, and explore the ability to "blindly" search for an optimal solution.

After training single-task baseline models, first step was to construct the loss for a multi-task model that included several common techniques for model performance enhancement. Chen et al. (2021) speaks to multiple different ways of constructing the loss, but given time constraints and with the understanding that we can address conflicting gradients elsewhere, we decided to use a simple linear combination of losses with equal weighting.

The second step involved finetuning the multitask model on the three datasets we had available. While there are an expansive array of MTL techniques (Chen et al., 2021) for sampling and training across different tasks, we started by applying a simplified round-robin training regimen which pulled together same-sized batches from each tasks. We further explored changing batch sizes (Stickland and Murray, 2019), and changing the number of epochs to execute training for shorter number of epochs to prevent over-fitting (Devlin et al., 2019). Next, we extended work by Bi et al. (2022) and Yu et al. (2020) to address conflicting gradients by projecting conflicting gradients on the normal plane. Additionally based on Stickland and Murray (2019) we employed annealed sampling which has been

shown to improve outcomes by oversampling smaller datasets towards latter epochs; enabling more generalized language representation that works better across tasks. Based on instabilities in the loss curve - evidencing exploding or vanishing gradients - we then implemented L2 regularization Hua et al. (2021) as a potential solution. Lastly we adjusted the task-heads by adding non-linear activation functions , and a combination of siamese linear layers and a cosine-similarity function (Reimers and Gurevych, 2019) to significantly enhance performance prior to the application of LASER.

With a "production" grade model in place and bench marked, we then adapted LASER as built by (Sharma et al., 2024) to reduce the rank and prune weight matrices post training as part of our pipeline. Keeping in mind their findings that the latter MLP layers are the most impactful across models - and our limited resources - we focused our exploration to systematically measuring the impact of individual and combined LASER interventions on MLP weights across the latter six layers of our BERT model.

4 Approach

BERT-based-uncased forms the starting point of this project, from which we systematically built single task models and a succession of multi-task models. Lastly, we built two LASER-enhanced models; one enhanced by first a single laser intervention, and the other to test a "blind" search method. Throughout we tested outcomes on three tasks to proof out value. Note that in addition to the below, we were considering hyper-parameter optimization trials through Bayesian optimization and had to cut the trial short due to lack of compute resources and the large body of evidence already available for such optimizations.

4.1 Base model

The single-task models adopt the architecture outlined in the default project handout () and include additional task-specific heads as shown in figure one below. Each single-task model includes a standard BERT-base-uncased model with scaled dot-product and multi-head attention. As a slight deviation from the BERT model found in Vaswani et al. (2023), we only use a single linear-add layer and GELU in the add norm. Note that for the purposes of this research we did not alter the number of layers or introduce new weight matrices. We used additive losses as shown in Stickland and Murray (2019) where the losses are constructed as $\mathcal{L}_{total} = \mathcal{L}_{SA} + \mathcal{L}_{PE} + \mathcal{L}_{STS}$. In the multiTask BERT model, all three classifier heads were trained alongside the model in simultanous batch training. We initialized the first Multitask BERT baseline with parallel training across all tasks. We cycled over smaller datasets multiple times in the initial version and kept batch sized the same. This formed our base MS-COMB baseline on which we could improve with various techniques.

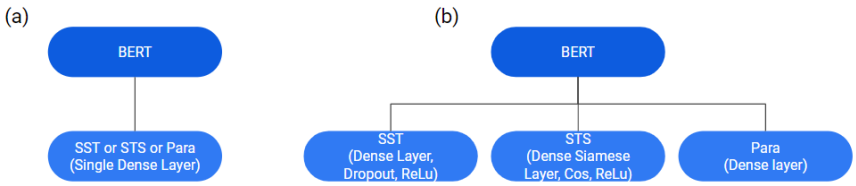

Figure 1: (a) The single-task BERT models included a single dense layer; over the course of experimentation we eventually landed on (b) the multitask BERT for which each classifier head included additional layers to create significant uplift on the model performance.

4.2 Gradient Surgery

The baseline MB-COMB model does not take into account conflicting gradients which can affect parallel task training. Yu et al. (2020) developed a tool called PCGrad to project the gradients of a task onto a conflicting task, removing the conflicting components of gradients and enabling progress in gradient descent. Specifically, gradient surgery involves taking gradient for each i'th *task* g_i, and projecting its gradient onto the normal plane of all other conflicting task gradients g_j: $g_i = g_i - \frac{(g_j \cdot g_i)}{\|g_j\|^2} \cdot g_j$. Adapting the best practices set up by Yu et al. (2020), we enabled random

selection of the i'th *task* task onto which other tasks would be projected in each training. This was shown in the original paper to improve training outcomes.

We further explored expanding on the tool provided by Yu et al. (2020), which flattens and concatenates all the gradients, by enabling weight by weight gradient surgery; by avoiding concatenation, we wanted to reduce the space for gradient surgery, avoid simplifying the complexity of calculating end-to-end gradients via simple concatenation. While the outcomes were of moderate improvement (see experiment section below), both the modified and the original algorithm significantly slowed training which was problematic for further experimentation on LASER given our compute limitations.

4.3 Annealed Sampling

Upon building the Gradient surgery toolset and realizing its downsides, we immediately explored the use of batch-level round-robin training with single batches from single tasks being chosen for each training iteration, but including the use of annealed sampling of each task. In annealed sampling, *tasks* included in a training iteration will be selected through a form of normalized probability sampling as identified in Stickland and Murray (2019) to reduce the risk of bias from different dataset sizes. Specifically the probability of a specific task's batch being chosen in any given iteration is given by: $p = \frac{N_i * \alpha}{\sum_i (N_i)}$ where $\alpha = 1 - 0.8 * \frac{e-1}{E-1}$ and e is the current epoch, and E is total number of Epochs. The original authors identified the need to sample tasks more evenly towards the end of training epochs even if it results in oversampling of smaller tasks. The method follows Stickland and Murray (2019)'s work for the most part, with some experimentation on the construction of the 0.8 constant in the construction of α.

4.4 Task-specific head design

The design of the individual classifier heads had evolved over time, we started with simple single linear layers to form the outputs as outlined in the default product handout. Based on we experimented with an adding an additional layers; settling on an additional linear layer on top of the original, and a added dropout layer and ReLu activation function between the linear layers. This helped drive up performance significantly. Additionally the STS task head was heavily modified given continued poor performance; with the model eventually landing on a dense siamese layer to train embeddings from both sentences, as well as a cosine similarity and RELU layer to add value. The choice of a RELU activiation function is unusual given that it should wipe out negative cosine values, but repeated experimentation with other configurations such as raw cosine output, and additional linear layers around the ReLu activation function provided poorer performance, and test-data seems to confirm the finding. Further analysis of the data itself may be warranted. The current working hypothesis is an imbalance in the training and devsets could make it more valuable to discern somewhat similar and very similar sentences. For the purposes of analyzing LASER, we left explorations of this space to a later date. The paraphrase extraction classifier was performing well from the initial setup and additional experimentation around the addition of ReLu activation and linear layers did not make significant improvements to the model.

4.5 Regularization of loss

Post the various changes made above, we noticed instability in the loss curves post the 3rd epoch, and as such chose to experiment with additional L2 regularization losses by modifying our Adam optimizer. After some experimentation where we exploeed $= [0.001, 0.01, 0.1, 1]$ we settled on $= 0.001$ as our chosen regularization constant. This provided a minor boost to performance on STS where there was some instability and degradation in performance in the latter epochs.

4.6 Layer Selective Rank Reduction

Post the inclusion of the various modifications above into a final MB-COMB-Final model, we began applying LASER interventions in a systematic way to explore the impact on model performance. LASER involves using Singular Value Decomposition (SVD) to alter select weight matrices across the transformer. As outlined in Sharma et al. (2024), given a matrix $W \in \mathbb{R}^{mxn}$ and $r \in \mathbb{N}$, a low-rank approximation is generating a $\widehat{W}$ that minimizes $\|W - \widehat{W}\|^2$ and meets $rank(\widehat{W}) \leq r$. The SVD of a matrix W is given by

$$W = U\Sigma V^\top$$

where

- $U = [u_1, u_2, \ldots, u_m] \in \mathbb{R}^{m \times m}$,
- $V = [v_1, v_2, \ldots, v_n] \in \mathbb{R}^{n \times n}$, and
- $\Sigma \in \mathbb{R}^{m \times n}$.

The column vectors of U and V are of $\mathbb{R}^m$ and $\mathbb{R}^n$ respectively, and Σ is a diagonal matrix whose diagonal entries are given by the singular values of W in descending order. Reducing rank involved eliminating only those components of Σ that are "higher ordered" or have smaller singular values; lower ordered components are preserved by LASER. A LASER intervention is characterized by three variables - parameter type τ which outlines the weight matrices of focus, layer number ℓ which described which layer the intervention occurs, and rank reduction ρ which describes the percentage of the maximum rank that should be preserved in the low rank approximation. The paper has useful visualization:
.

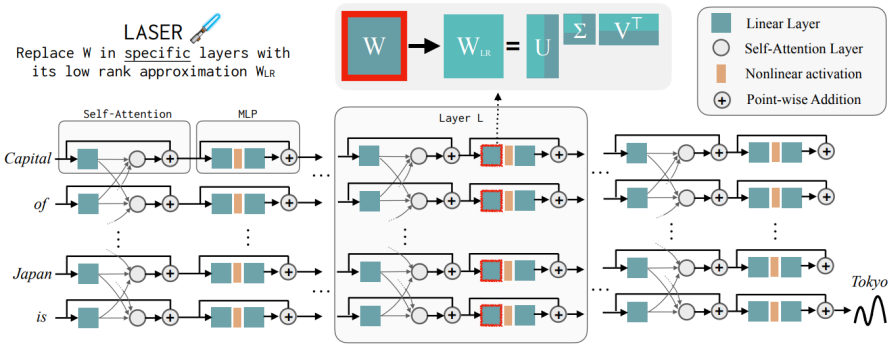

Figure 1: An example of LASER where weight W in layer L is replaced with WLR

In our experiments we utilize PyTorch's inbuilt SVD method, and apply interventions to MLP layers in BERT; specifically in layers 6,7,8,9,10 and 11, which Sharma et al. (2024) has identified as more likely to result in positive performance improvements in models like GPT-J and LLaMa 2 because they often encode conflicting but similar responses in high-ordered and low-ordered layers - resulting in a poor 'average' prediction. Our unique work here includes adapting the entire method for use in BERT and adapting the method for use in our pipeline.

Additionally, we heavily adapted/modified some code provided by Sharma et al. (2024) to reapply LASER multiple times to the same model; adapting it to selectively apply intervention in layers 6-12, with custom rank-preservation components, and focusing on both the interim.dense and output.dense weights. The goal was to have a narrower focus given computational budget limitations. Lastly we implement an alteration of the stated greedy-search algorithm presented in Sharma et al. (2024) to exhaustively search each layer for the most optimal intervention given intervention in the previously edited layer (starting again at Layer 12 and working backwards) Note we did limited trials to change batch size but saw some decline in performance in an unusual fashion and did not continue searching down that path due to resource constraints.

5 Experiments

We constructed several experimental models to help drive performance on three key tasks. Note we use acronyms in brackets going forward).

- Sentiment Analysis (SST). Tested on the Stanford Sentiment Treebank, this task involves classifying movie reviews from negative to positive
- Paraphrase Extraction (PARA). Tested on the Quora dataset, this tasks involves classifying two sentences as either paraphrases or not

- Sentence Textual Similarity (STS). Tested on the SemEval STS Benchmark dataset, this task ranks sentences on similarity.

5.1 Data

The datasets are specified within the project handout sheet. As per the handout the following datasets will be provided:

- SST: The data is sourced from the Stanford Sentiment Treebank. It contains 11,855 sentences which have been split into train (8,544 examples), dev (1,105 examples), and test (2,210 examples).The examples have been tagged with a sentiment score whose values range across 0 - negative, 1- somewhat negative, 2- neutral, 3- somewhat positive, and 4- positive.
- PARA: The data is sourced from the Quora Dataset. It contains 400,000 question pairs which have been split into train (141,506 examples), dev (20,215 examples), and test (40,431 examples). The examples have binary labels for paraphrase detection.
- STS: The data is sourced from the SemEval STS benchmark. It contains 8,628 different sentence pairs of varying similarity and labels, and have been split into train (6041 examples), dev (864 examples), and test (1726 examples) and is labeled on similarity on a scale from 0(unrelated) to 5 (equivalent meaning) Note the CFIMDB dataset (2,434 movie reviews with labels) was used for additional sentiment analysis in part 1 of the project but was not used in this part of the project. All datasets required some minimal pre-processing, tokenization and padding to ensure compatibility with BERT. We utilized CLS tokens to gain sentence level semantic representations.

5.2 Evaluation method

We use accuracy as our evaluation metric. It is defined as $Accuracy = \frac{ofcorrectpredictions}{oftotalpredictions}$. In our reporting here we show our accuracy results as derived from the dev set on each task. Furthermore we maintained a single set random seed across experiments such as to enable duplication of experiments.

5.3 Experimental details

All our reported experiments ran on common settings; 10 epochs, batch size of 8, fine-tuning learning rate of 1e-5, and a hidden layer drop out rate of 0.3. Note we also trained across all samples across all datasets and used resampling should any one dataset be exhausted early in the training process. Early experimentation along these lines were time consuming and were stopped early in favour of studying LASER. Note the LASER experiments were conducted on a V100 GPU while the previous experiments were done on a T4. Some experiments involved variations in key parameters that before they were finalized as mentioned above:

Gradient Surgery: At this early stage we varied batch size, epochs an even relative constituency of batches in parallel training of tasks. Specifically we tried batch sizes = [2,4,8,16,32]; in the end batch size of 8 was left as is due to lack of impact. Similarly we experimented with 3,4,8 and 10 epochs but left the default setting of 10. In the end default settings were maintained even after this particular branch of exploration was abandoned due to computational complexity:

Annealed Sampling: In this specific experiment we varied the definition of alpha; specifically the 0.8 constant defined in $\alpha = 1 - 0.8 * \frac{e-1}{E-1}$ to experiment with different batch constituencies. the values we tried include [0.6, 0.7, 0.8, 0.9]. In the end the 0.8 constant performed marginally better.

Loss Rates: In this specific experiment we varied $\lambda = [0.001, 0.01, 0.1, 1]$ and settled on 0.001 as mentioned previously.

Task Head design: In these specific experiments we varied the task-level dropout rate and tried [0.1,0.2,0.3,0.4] in the end 0.3 preformed better and was preserved- see performance below.

5.4 Results

To evaluate the efforts to build a production model and to evaluate LASER, use accuracy for SST and Para, and pearson correlation for STS. Table 1: Dev set accuracies of single-task models, multitask models, and LASER-enhanced models on the three fine-tuned tasks. The best results for each task is bolded.

Model	Accuracy (SST)	Accuracy (PARA)	P.Corr. (STS)
MB-Sst	0.517	0.509	-0.065
MB-PARA	0.136	0.788	-0.012
MB-STS	0.126	0.608	0.378
MB-COMB	0.506	0.779	0.360
MB-COMB-gs	0.493	0.786	0.380
MB-COMB-as	0.491	0.782	0.367
MB-COMB-csd	0.492	0.837	0.705
MB-COMB-final	0.508	0.842	0.705
MB-COMB-laser-l6-fcout*	**0.513**	0.842	0.700
MB-COMB-laser-l8-fcout*	0.503	**0.846**	0.694
MB-COMB-laser-l70-fcin*	0.487	0.843	**0.709**
MB-COMB-multilaser**	0.493	0.845	0.699

SST=Sentiment Analysis, PARA=Paraphrase Extraction, STS= semantic text similarity, COMB=multitask, the names for laser are the layer number and layer name where fcout and fcin are the out-dense and interim-dense weights, *one of 50 LASER-interventions, **single model with multiple interventions

Based on our findings we submitted the strongest single-laser intervention model to measure test-set performance and got SST(0.499) PARA (0.846) STS(0.676) with an overall test score of 0.728.

On the whole the quantitative results of MB-COMB and the individual laser interventions are not entirely surprising given the limited combination of interventions and layers that were included in this experiment.

However the multi-laser experiment which seeks to stack laser interventions was surprisingly underperforming expectations. Constructing a simple score adding the evaluation metrics of all three tasks, we found 48 laser interventions that exceeded MB-COMB in various combinations. We hypothesize that performance did not add up as a result of two factors; first the overall performance is probably low because of layer-to-layer interactions removing the impact of individual layer-wise impact. Secondly the way in which we constructed the north star metric (sum of individual metrics) to choose each model may leave the algorithm susceptible to going down sub-optimal paths.

6 Analysis

6.1 Qualitative Deepdive

On the whole the system worked somewhat as expected given the state-of-the art across all three tasks; the SST task in particular is a difficult one to improve with LASER without significant further experimentation.

Sentiment Classification (SST). A visual inspection of the SST dataset showcases a series of reviews that involve use of unusal phrasing that make up the middle-scored reviews (scores of 2-3). For example "Meyjes ' provocative film might be called an example of the haphazardness of evil" may be seen as either positive or negative by a human reviewer, but model may latch onto specific terms like evil and give it a negative review. Despite various attempts to improve the model with LASER interventions it seems like there's a natural ceiling to the signals that can be extracted from this dataset.

Paraphrase Extraction (PARA) and Semantic Textual Similarity. High performance on this set (and eventually on STS) is not unexpected given the sheer size of the training data, and the similarity between the tasks. The big difference in performance was the use of the cosine similarity layer and the use of a ReLu function; we did not have the chance to examine the data further due to time constraints, but we suspect that one reason why the ReLu function worked was due to bias in the training dataset; the model may have been better able to differentiate between the somewhat similar

and very similar but had difficulty with the other end of the spectrum. On further investigation we would look into the datasets deeper.

LASER The underperformance of LASER was intriguing considering the large impact it had on experiments like GPT-J and LLaMa 2 in previous experiments. As mentioned layer-to-layer interaction may play a much larger role than expected.

7 Conclusion

The main finding is from this exploration is LASER while exciting in its initial vision and scope, may need further studies in it's applications to smaller models or multi-task environments. Performance improvements were marginal but future explorations on stacking LASER interventions should focus on hyper-parameter style search of the intervention space - focusing on Bayesian Optimization methods given the clear patterns in accuracy and loss across layers.

References

Tamay Besiroglu, Sage Andrus Bergerson, Amelia Michael, Lennart Heim, Xueyun Luo, and Neil Thompson. 2024. The compute divide in machine learning: A threat to academic contribution and scrutiny?

Qiwei Bi, Jian Li, Lifeng Shang, Xin Jiang, Qun Liu, and Hanfang Yang. 2022. MTRec: Multi-task learning over BERT for news recommendation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2663–2669, Dublin, Ireland. Association for Computational Linguistics.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2021. Understanding and overcoming the challenges of efficient transformer quantization.

Shijie Chen, Yu Zhang, and Qiang Yang. 2021. Multi-task learning in natural language processing: An overview.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jonathan Frankle and Michael Carbin. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv: Learning*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Hang Hua, Xingjian Li, Dejing Dou, Cheng-Zhong Xu, and Jiebo Luo. 2021. Noise stability regularization for improving bert fine-tuning.

Woosuk Kwon, Sehoon Kim, Michael W Mahoney, Joseph Hassoun, Kurt Keutzer, and Amir Gholami. 2022. A fast post-training pruning framework for transformers. In *Advances in Neural Information Processing Systems*, volume 35, pages 24101–24116. Curran Associates, Inc.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023. Locating and editing factual associations in gpt.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Pratyusha Sharma, Jordan T. Ash, and Dipendra Misra. 2024. The truth is in there: Improving reasoning with layer-selective rank reduction. In *The Twelfth International Conference on Learning Representations*.

Asa Cooper Stickland and Iain Murray. 2019. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Huanrui Yang, Minxue Tang, Wei Wen, Feng Yan, Daniel Hu, Ang Li, Hai Li, and Yiran Chen. 2020. Learning low-rank deep neural networks via singular vector orthogonality regularization and singular value sparsification.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning.

Xiyu Yu, Tongliang Liu, Xinchao Wang, and Dacheng Tao. 2017. On compressing deep models by low rank and sparse decomposition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7370–7379.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2023. A survey on model compression for large language models.